CMPSC 465 Data Structures & Algorithms
Spring 2024 Mehrdad Mahdavi and David Koslicki **HW 3**

Due February 1, 10:00 pm

Instructions: You are encouraged to solve the problem sets on your own, or in groups of up to five people, but you must write your solutions strictly by yourself. You must explicitly acknowledge in your write-up all your collaborators, as well as any books, papers, web pages, etc. you got ideas from.

Formatting: Each problem should begin on a new page. Each page should be clearly labeled with the problem number. The pages of your homework submissions must be in order. You risk receiving no credit for it if you do not adhere to these guidelines.

Late homework will not be accepted. Please, do not ask for extensions since we will provide solutions shortly after the due date. Remember that we will drop your lowest two scores.

This homework is due Thursday, February 1, 10:00 pm electronically. You need to submit it via Gradescope (Class code XX7RVV). Please ask on Campuswire about any details concerning Gradescope and formatting.

1. (20 pts.) **Heap and Heapsort.**

(a) Run Build-Heap on the array $[12,10,4,7,6,1,3,0,8,5]$ to construct a min heap. Write down the resulting array. Then, run the loop in Heapsort for three iterations. For each iteration write down the array after Heapify-Down is finished. You do not need to show intermediate steps.

(b) Show that the leaves of a heap occupy the positions $\lfloor \frac{n}{2} \rfloor, \ldots, n-1$ of the array.

(c) In the procedure Build-Heap, why do we decrease i from $\lfloor \frac{n}{2} \rfloor - 1$ to 0 instead of increasing from 0 to $\lfloor \frac{n}{2} \rfloor - 1$? Give an example array where increasing i fails to create a valid min heap.

2. (15 pts.) **Sorting Lower Bound.** Show that any array of integers $x[1 \ldots n]$ can be sorted in $O(n+M)$ time, where

$$M = \max_i x_i - \min_i x_i$$

For small M, this is linear time: why doesn't the $\Omega(n \log n)$ lower bound apply in this case?

3. (15 pts.) **Binary search.**

(a) Suppose we want to check if a sorted sequence A contains an element v. For this, we can use Binary Search. Binary Search compares the value at the midpoint of the sequence A with v and eliminates half of the sequence from further consideration. The Binary Search algorithm repeats this procedure, halving the size of the remaining portion of the sequence each time. Write a recurrence for the running time of Binary search and solve this recurrence.

(b) Ternary Search is a generalization of Binary Search that can be used to find an element in an array. It divides the array with n elements into three parts and determines, with two comparisons, which part may contain the value we are searching for. For instance, initially, the array is divided into three thirds by taking $mid1 = \lfloor \frac{n-1}{3} \rfloor$ and $mid2 = \lfloor \frac{2(n-1)}{3} \rfloor$. Write a recurrence for the running time of Ternary search and solve this recurrence.

4. (20 pts.) **Counting number of inversions.** We are given a sequence of n distinct numbers $a_1, \ldots, a_n$. We define an inversion to be a pair $i < j$ such that $a_i > a_j$. Give an $O(n \log n)$ algorithm to count the number of inversions. (*Hint: Consider modifying Merge Sort to return both the sorted array and the number of inversions. Use recursion to sort and count the inversions of the two halves. Then, during the merging process, count the number of inversions needed for the current recursive step.*)

5. (30 pts.) **Median Heaps.** Consider an array of N numbers $[x_1, x_2, \ldots, x_N]$ that you will be receiving one-by-one in a single pass (i.e., as a stream of numbers). You are not allowed to re-query previous numbers. Design an effective data structure involving one or more heaps that, after receiving the n-th number, reports the median of the numbers $x_1, x_2, \ldots, x_n$, observed so far. The time complexity of your algorithm should be $O(\log n)$ per number received in the worst case. For simplicity, assume there are no duplicates in the stream. Explain the algorithm/data structure, running time, and correctness.

Hint: If n is odd, the median of the data stream $[x_1, x_2, \ldots, x_n]$ is the middle element of the sorted data stream, else the median is the average of the middle two elements of the sorted data stream.

6. (0 pts.) **Acknowledgments.** The assignment will receive a 0 if this question is not answered.
(a) If you worked in a group, list the members of the group. Otherwise, write "I did not work in a group."
(b) If you received significant ideas about the HW solutions from anyone not in your group, list their names here. Otherwise, write "I did not consult with anyone other than my group members."
(c) List any resources besides the course material that you consulted in order to solve the material. If you did not consult anything, write "I did not consult any non-class materials."